Semiannual Technical Summary

Information Processing Techniques Program

Volume I: Packet Speech/Acoustic Convolvers

31 December 1974

Prepared for the Advanced Research Projects Agency
under Electronic Systems Division Contract F19628-73-C-0002 by

Lincoln Laboratory

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

LEXINGTON, MASSACHUSETTS

The work reported in this document was performed at Lincoln Laboratory, a center for research operated by Massachusetts Institute of Technology. This work was sponsored by the Advanced Research Projects Agency of the Department of Defense under Air Force Contract F19628-73-C-0002 (ARPA Orders 2006 and 292:).

This technical report has been reviewed and is approved for publication.

FOR THE COMMANDER

Eugene C. Raabe

Eugene C. Raabe, Lt. Col., USAF
Chief, ESD Lincoln Laboratory Project Office

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

LINCOLN LABORATORY

INFORMATION PROCESSING TECHNIQUES PROGRAM
VOLUME I: PACKET SPEECH/ACOUSTIC CONVOLVERS

SEMIANNUAL TECHNICAL SUMMARY REPORT
TO THE
ADVANCED RESEARCH PROJECTS AGENCY

1 JULY - 31 DECEMBER 1974

ISSUED 14 APRIL 1975

LEXINGTON MASSACHUSETTS

ABSTRACT

The Information Processing Techniques Program sponsored by DARPA at Lincoln Laboratory consists of three efforts: Packet Speech (Network Speech Compression), Acoustic Convolvers, and Airborne Command and Control. In this Semiannual Technical Summary, the first two areas are reported in Vol. I and the third in Vol. II. In addition, Vol. I contains a brief summary report on work in Speech Understanding completed in FY 1974.

CONTENTS

INFORMATION PROCESSING TECHNIQUES PROGRAM

I. PACKET SPEECH

A. OVERALL AIM OF PACKET SPEECH PROGRAM

The long-range objective of the Packet Speech Program is to develop and demonstrate techniques for efficient digital speech communication on networks suitable for both voice and data, leading eventually to better interaction between people and computers in a multicomputer network environment. The ARPA Network has demonstrated the advantages of message-switched data networks. Presently, the ARPA-sponsored Network Speech Compression (NSC) group is studying the effects of the network on digital speech transmission. Ultimately, development of a successful voice-data network will require application of many of the results of the past half-century of speech research (drawing upon such topics as speech recognition, speech synthesis, speech bandwidth compression, speaker authentication) for effective digital speech communication. Current efforts of the NSC group are strongly oriented toward adaptation of the ARPANET to fulfill voice needs and the problems of speech bandwidth compression. At present, this effort is strongly cooperative and should lead, toward the end of FY 75, to an NSC experimental voice communication system plus more information on the effects on digital speech of packet technology.

B. ACTIVITIES OF THE NSC GROUP

In addition to Lincoln Laboratory, the present members of the NSC group comprise: Information Sciences Institute; Bolt, Beranek and Newman; Speech Communications Research Lab; Culler-Harrison Incorporated; Stanford Research Institute; University of Utah; and M.I.T. Both Utah and M.I.T. have small research efforts which, at the moment, are peripheral toward the major development goal of the NSC group (i.e., establishing a speech system and then a conferencing capability). What follows is a review of those aspects of the work of ISI, BBN, SCRL, CHI and SRI which impinge on the Lincoln effort.

ISI has worked closely with Lincoln in setting up the first ARPANET two-way voice system; this is described in some detail in Sec. I-F. Communications were established via a Lincoln fast digital processor (FDP) simulation and an ISI SPS-41 simulation of a continuously variable slope delta (CVSD) modulation system, which was described in our previous SATS.* The network proved able to accommodate no more than 10 kb/s, a rate at which CVSD was reasonably intelligible but not of satisfactory quality. These experiments have been very useful in giving us a "feel" for the network voice capability and the "glitches" that occurred. Presently, ISI and Lincoln are planning a first conferencing experiment, using the CVSD hardware purchased for ARPA by Lincoln. In addition, ISI is in the process of programming the SPS-41 for real-time implementation of the linear predictive coding (LPC) algorithm.

BBN, SCRL, and SRI also are preparing to use the SPS-41 as a real-time speech processor. SCRL has been given the responsibility of establishing the precise details of the algorithm. ISI is preparing the voice protocol for the network. SCRL also has responsibility for the ELF

*Speech Understanding Systems Semiannual Technical Summary, Lincoln Laboratory, M.I.T. (31 May 1974), p. 12, DDC AD-783284/3.

program which will constitute the common operating system for the NSC group. It is interesting to note that ISI, SRI, SCRL, BBN, and Lincoln all will have the same facility, namely, a PDP-11/45 or -11/40, so that software compatibility should be very good.

C. SUMMARY OF LINCOLN'S ACTIVITIES IN THE NSC GROUP

Lincoln's role in the NSC group has stressed the experimental aspect of ARPANET voice communication. To this end, we have participated in a 10-kb/s CVSD voice experiment with ISI, as mentioned previously, and are now involved in an LPC voice experiment with CHI. In addition, we have procured 12 CVSD processors to set up a conferencing strategy experiment with ISI. We will participate in the NSC packet-voice network which should be assembled early in 1975. Finally, we have run a series of "fake-host" experiments to discover more about network delays and throughput.

We also are active in the speech compression area, as described in Secs. I-G and I-H. Our real-time FDP facility has allowed us to evaluate large quantities of statistical data on the coding of the reflection coefficients, which are the parameters of the LPC-derived vocal tract. Based on these measurements, we are devising a strategy for efficient coding. We also are studying and experimenting with variable-rate LPC and, finally, we are studying methods other than pitch extraction of exciting the LPC synthesizer.

D. CVSD HARDWARE DESIGN AND PROCUREMENT

At the end of calendar 1974, Lincoln was asked to study the design and construction of a simple state-of-the-art speech waveform encoder in the 10- to 20-kb/s range. These encoders were to be procured from an outside contractor according to a Lincoln Laboratory technical specification, and used as an early digital voice input to the ARPANET. At the time, the only viable waveform encoding schemes were the adaptive differential pulse code modulation (ADPCM) approach taken by the Bell Telephone Laboratories for efficient storage and playback of speech with computers, and the (CVSD) modulation approach taken by several DOD contractors.

To decide between ADPCM and CVSDM, a real-time simulation of each encoding technique was programmed on the FDP, and the processed speech was evaluated by "talking through" each simulation as a speech encoder-decoder. It was clear that ADPCM at 18 kb/s (6-kHz Nyquist sampling and 3 bits per sample) was less noisy and somewhat better quality than CVSD at 18 kb/s (1-bit delta signal at 18-kHz oversampling of 3-kHz band). However, ADPCM could go no lower in rate without compromising the input speech bandwidth or going to an unacceptable 2 bits per sample, while CVSDM could be lowered in rate continuously by varying the sampling clock. This was an important consideration for flexible ARPANET use. In fact, CVSDM is useable (at low quality) down to 8 kb/s and is used on the network most commonly at 10 kb/s. The output of an ADPCM coder is a PCM word of 3 bits and requires some transmitted synchronization information for framing if used in a serial transmission. A CVSD output word is a single-bit delta code and requires no synchronization or framing consideration for serial transmission.

The CVSDM algorithm is a very simple one requiring a minimal one- or two-small-circuit-board realization. The ADPCM algorithm realization needed some redesign to reduce it to an equivalent two-circuit-board format. This was done, and the prototype and design and layout were available for contractor use.

After listening to prototypes of each device, and weighing the above factors with ARPANET use in mind, it was decided to procure CVSD equipments for NET use.

It was decided not to incorporate packetizing and silence thresholding in the encoder-decoder because of the experimental nature of these equipments and the need for flexibility in silence and packetizing parameters. This flexibility would be available in host software, rather than speech-encoder hardware.

The final procurement design consisted then of a CVSDM encoder and decoder, handset and audio circuits, crystal clock and divider circuits, parallel interfaces to PDP-11 unibus, serial interfaces to modems, and power supplies. The device can operate at fixed serial and parallel modes of 18, 16, 14, 12, 10 and 8 kb/s.

A Request for Quote was issued to several contractors in April 1974 and Magnavox-General Atronics, Philadelphia was chosen to supply 12 of these devices by 31 December 1974 to ARPA (see ARPA NSC Note No 15, 23 April 1974).

Present plans call for five units at ISI and five at Lincoln Laboratory for use in conferencing experiments at each site, as well as for inter-site use.

E. NETWORK SPEECH EXPERIMENTS

1. Introduction

A series of experiments has been planned to focus the NSC effort. To date, two experiments have been carried out involving actual speech communication between two sites via the ARPANET. Other experiments involving only one site have been undertaken to measure network characteristics and to demonstrate the effect of network delays on speech communication. Each experiment involves the implementation of programs to simulate in real time the vocoding technique being explored, to handle an appropriate protocol for the ARPANET communications, and to measure and record pertinent information during the operation of the programs. The expected results are both subjective evaluations and detailed measurements which can yield insights into improvements in both vocoding techniques and network characteristics.

All our experiments have made use of TX-2 and the FDP, with the FDP handling the vocoder simulation and TX-2 managing network protocols, data formatting, and measurement recording. Communication between TX-2 and the FDP makes use of the 1.6-Mb/s serial lines installed for speech understanding research use.

The results to date have shown that speech of acceptable quality can be transmitted through the ARPANET with little difficulty. However, the network in its present form introduces unacceptable delays into conversations. These delays have a base component due to the store-and-forward process which moves a message from node to node, plus a variable component due to other network traffic and the action of network control algorithms. The resulting dispersion of network message delays requires that the receiver of a speech data stream introduce a further smoothing delay to avoid "glitches" caused by the network falling behind the speech output process.

A major result of the experiments to date has been a better understanding of network delays and the mechanisms in the net which cause them. BBN, the ARPANET principal contractor, has followed the experiments closely and has undertaken to install a series of changes in network control algorithms. These changes, plus the introduction of an additional 50-kb/s line in

the Los Angeles area, should greatly improve the prospect for ARPANET speech communication among the NSC participants.

The following sections discuss particular experiments in more detail.

2. CVSD Experiment

The first experiment made use of the CVSD vocoding technique discussed in the previous SATS.* This technique was chosen for the first experiment because the programs to simulate it are relatively simple and easily implemented. It can be used at any bit rate, but quality deteriorates noticeably below about 16 kb/s.

The experiment was carried out between Lincoln and ISI, with the ISI implementation carried out on their PDP-11/SPS-41 facility. A protocol was agreed upon which used control messages to establish and break connections and data messages to transmit the CVSD bit stream during times when the speaker was actually talking. When the silence detection in the CVSD algorithm* first indicated silence after a "talkspurt," a special silence message was sent on the net and no further data messages were transmitted until talking resumed. The protocol called for special silence messages to be sent at regular intervals during a long silence to inform the receiver that the transmitter was still alive and well. By not transmitting the CVSD bit stream during silences, the average data rate was reduced and the receiver could recover from network delays which had accumulated during intervening talkspurts.

The ARPANET flow control algorithms are designed to give fast delivery and low average throughput for short messages, and slow delivery but high throughput for long (multipacket) messages. Unfortunately, the CVSD bit stream required a high throughput and forced the use of long messages with their associated slow delivery. The slow delivery results because the network will not begin transmitting a multipacket message until it has ascertained that buffer space for the message is available in the destination IMP. To reserve the necessary space, the source and destination IMPs must exchange control messages. Once reservations have been set up for the four messages which the network will allow to be in the system at one time between any source-destination pair, the reservations will be held implicitly and data will move smoothly without further delay. Unfortunately, the delay at the beginning of a talkspurt delays the entire talkspurt and there is no way to take advantage of the fact that the network performance improves after the initial reservation delay.

In the CVSD experiment, buffer management considerations in the ISI PDP-11 placed a limit on message length of two packets, which resulted in just under 2000 bits of data per message. With that size message, it was observed that the network could sustain only about 10-kb/s average throughput between Lincoln and ISI. With the CVSD speech adjusted to 10 kb/s, successful speech communication was achieved in early September 1974.

While the CVSD experiment demonstrated that speech communication via the ARPANET could be achieved, performance was far from satisfactory. At 10 kb/s, the CVSD speech was below an acceptable quality level, and the network delays were excessive. Further experimentation with the high-rate CVSD technique has been deferred until proposed changes in the ARPANET have been carried out. These changes should allow increased throughput with reduced delays.

*See pp. 12-14 in previous SATS (31 May 1974).

3. LPC Experiment

The second experiment utilized the LPC vocoding technique. The plans for LPC speech communication eventually will involve most of the NSC sites, but CHI and Lincoln have the first running systems capable of handling the LPC algorithms and have therefore carried out the first tests of LPC speech on the net. Since there are many possible variations of the LPC algorithm and many ways to code the parameters for transmission, a more complex protocol was designed by D. Cohen of ISI and approved by the NSC group. This Network Voice Protocol (NVP) allows the parties to negotiate the details of the conventions to be used in each particular session prior to establishing the data connections.

For the CHI-Lincoln LPC experiment, a subset of the NVP was implemented which allowed only one particular LPC coding technique to be used. This coding resulted in a bit rate of 3450 b/s during talkspurts. Silences were handled in a fashion similar to the CVSD experiment, with special messages at the start of a silence and at intervals thereafter.

The data rate of 3450 b/s allows the use of single-packet messages in the net which avoids the reservation delays observed in the CVSD experiment. By the last week in November 1974, when the first tests were carried out, some of the network changes being made by BBN had been completed, and network performance was somewhat improved over that which was observed at the time of the CVSD experiment.

The results of the first LPC experiment were quite encouraging. LPC speech at 3450 b/s is of much higher quality than CVSD speech at 10 kb/s. Network delays for the conditions of the first LPC tests were longer than one would desire, but we are confident that significant improvements are likely with the completion of the planned network changes and refinements of our coding and transmission algorithms. Even in its present form, LPC speech via the ARPANET could be used without significant annoyance due to glitches, delays, or overall quality.

4. "Fake-Host" Experiments

In order to be able to explore possible interactions between vocoding techniques and network delay and glitch phenomena without having to bother people at some other NSC site, we have planned and partially implemented software to allow us to have both ends of a network conversation at Lincoln. This software makes use of a feature built into the IMP software which allows messages to be sent to a particular fake host at any IMP. Such a message is discarded and a Request for Next Message (RFNM) is returned to the sender. RFNMs from fake-host transmission have the same delay characteristics as RFNMs from real-host transmissions.

In our fake-host experiments, we keep the vocoded data stream in TX-2 and send dummy messages of appropriate length and at appropriate times to a fake host at some IMP. We interpret the return of the RFNM as being equivalent to the satisfactory arrival of the message corresponding to the RFNM. We then allow the vocoded data corresponding to the message to be played out to our local listener. This technique introduces a little more delay than would be seen by a real host at the IMP to which we sent the message for discard.

To date, we have implemented a CVSD fake-host program and expect to have an LPC fake-host capability early in 1975. Our current implementation on TX-2 and the FDP allows for vocoding only one party in a conversation. When the PDP-11/SPS-41 system becomes available, we could provide similar programs for that facility and be able to vocode both parties in a full-duplex fake-host conversation.

F. NETWORK MEASUREMENTS

We have developed a program on TX-2 for carrying out delay and throughput measurements on the ARPANET. The program allows for convenient variation of measurement parameters, and convenient display and analysis of results. The basic delay measurements are made by transmitting in fake-host mode and determining RFNM delays. Some of the results already obtained have pinpointed difficulties in speech transmission on the ARPANET, and helped focus efforts in modifying the network to alleviate these problems. The measurement program is a convenient tool for monitoring the effect of network changes.

1. Input Parameters for Network Measurements

The key parameters which can be varied in carrying out our measurements are (a) speech source rate (bits/sec), (b) message size (bits), (c) lengths of data bursts and silence intervals, (d) destination HOST, and (e) time of day.

The speech source rate in bits/second (b/s) and the message size in bits are the two parameters which describe the network input during nonsilence intervals. During silence, we assume that nothing is sent. For example, to simulate CVSD transmission at 10,000 b/s to SCRL, we might transmit messages with 1250 data bits to SCRL every 125 msec, using the fake-host protocol. We can simulate the effects of silence detection by sending variable-length bursts of data, separated by silences. By using the fake-host feature, we can investigate the effect of transmitting to various points in the net, over various numbers of hops. Another important variable in the experiment is network traffic, and we can carry out measurements at various times of day to probe peak and light traffic loads.

2. Parameters to be Measured

For each message we transmit, four key times can be determined, namely: T0, the time when the message is filled by the speech source and is ready for transmission; T1, the time when this message begins to leave the HOST; T2, the time when the HOST-IMP transmission is completed; and T3, the time of arrival of the RFNM. If things are working well, the dominant delay would be the round-trip transit delay D3 = T3 – T2. But D2 = T2 – T1 could be appreciable due to delays in obtaining a space allocation at the destination IMP, and D1 = T1 – T0 represents a queueing delay in the source HOST which could build up if the network throughput fell behind the input source rate for a period of time.

There is another time, the arrival time Ta of the message at the destination HOST, which is of great interest but cannot be measured with fake-host techniques. The delay actually experienced by a particular message is not Dr = T3 – T0, but rather Da = Ta – T0. However, if our message size is significantly larger than a RFNM message size (160 bits), then the time for the RFNM to return should be much smaller than the data message transit time, and our measured delay Dr will be only a slight overbound on the actual network transit delay. In addition, the RFNM transit time should have much less variation from message to message than the data message transit time.

For each given set of input parameters as described above, we collect histories of the four pertinent time measurements described above. These data can be manipulated and displayed

conveniently. For example, histograms of delays can be plotted easily. Transient effects caused by leaving and entering silence intervals can be investigated. Steady-state average delays for given source rates can be determined. From these data, it is possible to gain insight into the buffering strategies needed at both the transmitting and receiving HOSTs. By distinguishing between the delays D2 and D3, it is possible to some extent to separate flow-control delays from message-transit-time delays. Finally, we can investigate the maximum source rate that can be transmitted without an indefinite buildup of delay.

3. Measurement Example

An example of data collected with our measurement program is shown in Fig. I-1. The input parameters for these data are: (a) speech source rate 10,000 b/s; (b) message size 1667 bits (2-packet messages at 6 messages/sec); (c) 15-sec data bursts separated by 4-sec silence periods; (d) transmission from Lincoln to ISI (10 hops in the ARPANET); (e) mid-day traffic conditions. The data are quite typical of the delays we encountered in talking CVSD with ISI at 10 kb/s. In order to achieve the 10-kb/s throughput, it was necessary to transmit multi-packet messages, with the resulting disadvantage of requiring buffer reservations at the destination HOST before beginning to transmit.

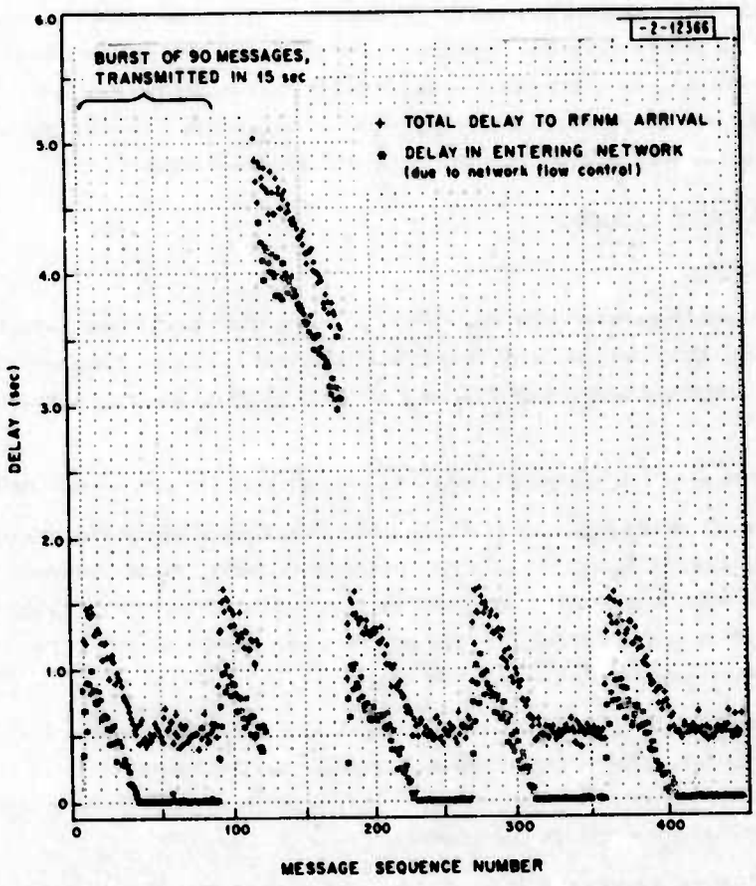


Fig. I-1. Two-packet messages to ISI (10 hops) at 10 kb/s.

Let us focus attention on the first burst of data in Fig. 1-1, representing 15 sec of transmission at 6 messages/sec. The lower set of points represents the time (T2 – T0) spent by each message waiting in the source host before entering the network, and the upper set of points represents the total delay Dr, including network transit delay for the message and RFNM. Initially, a large transient delay builds up as we wait for storage allocation at the destination. Before we get the full required allocation, a backlog of messages builds up and it takes about 6 sec before we can work off this backlog to the point where only the network transit delays are seen. Note that this transient will occur every time we start talking after silence, since we want to send no data during silence and therefore lose our storage allocation. The several bursts shown are all quite similar except for the second one, where the situation is aggravated due to a "glitch," or unusually large delay. We have observed similar glitches under other measurement conditions, and have brought our observations to the attention of BBN (the network systems contractor) who has introduced some network modifications which seem to substantially reduce the frequency of these glitches.

An important point to remember is that it is the peak delay, not the average or the minimum, which is relevant for speech transmission. Once we get 1.5 sec behind, as in this example, we will stay that far behind until the next silence period, since the speech is played out at a uniform rate.

This example was intended primarily to illustrate the measurements capability, and not to indicate present or future network capabilities (in fact, network changes have been made since these data were taken). In particular, it is probable that in the near future, 10-kb/s transmission over 10 hops will be possible with single-packet messages, so that the transient delays illustrated in the example will not occur at this transmission rate.

G. VARIABLE-RATE CODING

1. Introduction

In a packet-speech communications network, there is no particular advantage in having a constant data rate; thus, useful reduction in average traffic can be realized via variable-rate speech coders. Itemized below are different methods of implementing a variable-rate speech processor:

(a) Detection of silence intervals and cessation of transmission during these intervals.

(b) Use of coding algorithms which take advantage of temporal fluctuations in the instantaneous "natural" speech rate: for example, during voiceless sounds, fewer bits may be needed to reproduce perceptually satisfactory synthetic speech. Another example: during steady portions of voiced sounds, parameters of the speech may be transmitted at a lower rate.

(c) Adaptation to the statistics of a particular speaker's parameters: in this scheme, the speech processor would be compiling a running short-time statistic. When the two statistics are close, fewer quantization levels would be required to accurately represent the speech parameters.

Experiments using a conjunction of all three techniques resulted in surprisingly good LPC speech at an average rate of 1400 b/s. In a packet-speech network, assuming that for a two-way conversation only one speaker at a time is talking, this implies an average rate of 700 b/s per conversation.

Throughout our investigation, the analyzer and synthesizer remained unchanged. Analog pre-emphasized speech (6 dB/octave, 300 to 3000 Hz) was sampled at 130-μsec intervals and quantized to 12 bits. The signal was then windowed (19.5-msec Hamming window). The K-parameters were computed via Levinson's recursion with 18-bit fixed-point arithmetic. Pitch was determined by the Gold-Rabiner pitch detector. The synthesizer used the acoustic tube representation, and parameters of the synthesizer were interpolated every 5 msec. The frame rate was 19.5 msec.

It should be noted that our definition of the K's differs from the ARPA standard by a sign bit (e.g., $K_{0.ours} = -K_{0.ARPA}$). Also, our indexing scheme goes from K_0 to K_9, instead of from K_1 to K_{10}.

2. Histograms of the K-Parameters

Because we were interested in the behavior of the K-parameters under various constraints, we decided to collect data on the distribution of each K. Histograms were collected for K's (0 through 9) under the following categories: (a) over all speech, (b) voiced speech only, (c) unvoiced speech only, and (d) silence only. Frames in which the energy was below a threshold were considered silence. The remaining frames were categorized as voiced or unvoiced by the pitch detector. Figures I-2 and I-3 show some results for the combined data on 7 speakers (4 male and 3 female) over 25 minutes of speech, recorded through a handset.

The following points can be made about the histograms:

(a) There is a large peak in K_0 near 0 and in K_1 near +0.5 which is due almost exclusively to silence, and therefore can be ignored.

(b) Ninety percent of the time voiced K_0 is above 0.5, and is less than 0 only 2.5 percent of the time. Unvoiced K_0 spreads over the entire region from −1 to +1.

(c) The histogram for voiced K_1 is an asymmetric function which peaks near −1, whereas unvoiced K_1 exhibits a symmetric histogram which peaks at 0.

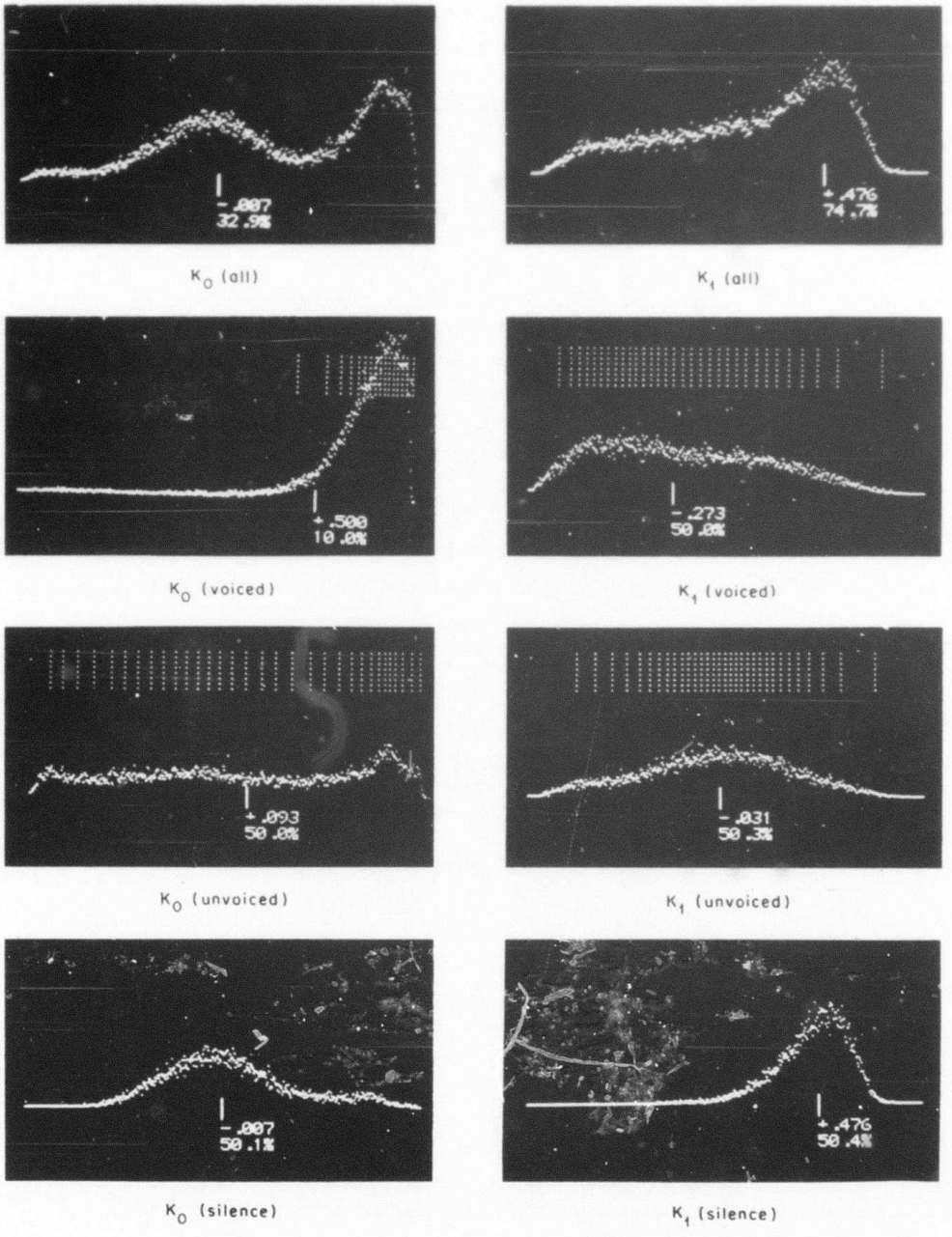
(d) For the high-order K's (2 through 9) the histograms are sharpest for K_9 and broadest for K_2, and the unvoiced histograms are sharper in general than the voiced counterparts and also are centered at a different location.

3. Equal-Area Coding Scheme

Given the above results, we concluded that arcsine tables from −1 to +1 were extremely wasteful for voiced K_0 and K_1, and that even truncated arcsine (±0.7) was wasteful for some of the higher-order K's. In addition, the distribution was sufficiently different between a voiced K histogram and the unvoiced counterpart, and between one voiced K and another, to warrant separate tables for each K and for voiced vs unvoiced speech.

We decided to try a well-known coding method which has been used frequently in the past for parameter encoding. We divided up the histogram of each K into equal-area regions, and one level was allocated to each region. A separate table was used for each voiced K and for each unvoiced K (20 tables in all). For silence frames, the value of all the K's was set to 0.

While 20 tables may sound overwhelming, at the low bit rates that we are using only 248 decimal locations of memory are required to store all the voiced tables for both coding and decoding. This number is comparable to the amount of memory required for the present ARPA standard arcsine tables.



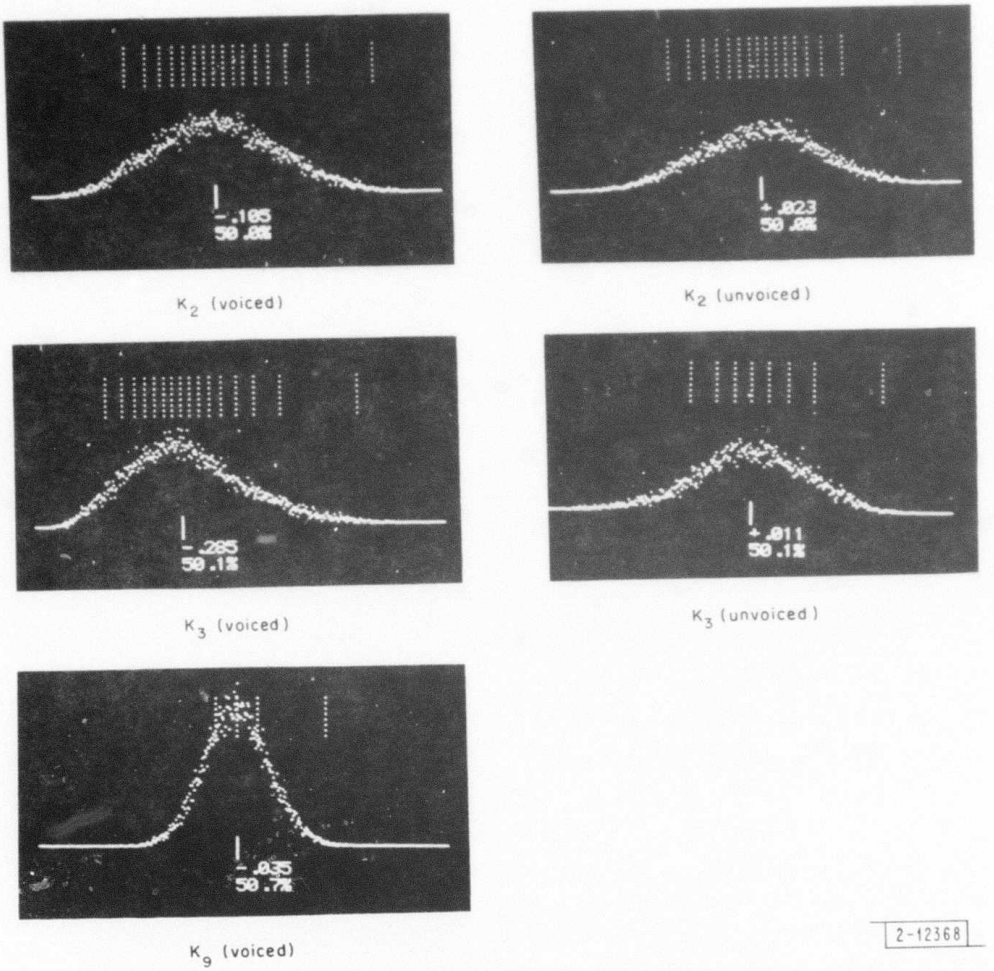
K_0 (all) K_1 (all)

K_0 (voiced) K_1 (voiced)

K_0 (unvoiced) K_1 (unvoiced)

K_0 (silence) K_1 (silence)

Fig. I-2. Histograms of LPC parameters.



K_2 (voiced)

K_2 (unvoiced)

K_3 (voiced)

K_3 (unvoiced)

K_9 (voiced)

Fig. I-3. Comparison of voiced and unvoiced LPC parameter histograms.

The experimental system was such that the histogram data were collected in real time, and the tables could be updated periodically to include the most recent data. Also, at any time, the histogram buffer could be cleared and new tables could be determined based on only the most recent data. The number of bits allocated to each K parameter was a variable which could be adjusted on-line, and different numbers could be allocated to voiced vs unvoiced K's.

Using the experimental system, we were able to determine an optimal bit allocation scheme. We found that only 4 bits were needed for voiced K_0, whereas unvoiced K_0 required 5 (because its histogram has a much larger spread). We decided empirically through listening tests and through observations of the histogram distribution that a lower-limit bit-allocation scheme, as follows, achieved reasonably good quality speech:

Voiced	4 5 4 4 3 3 3 3 3 2 (K's 0 through 9)
Unvoiced	5 5 4 3 2 1 0 0 0 0

It is interesting to note that voiced K_1 requires more bits than voiced K_0. Also, we found that for unvoiced frames the 4 highest-order K's could be fixed at their average value, suggesting that unvoiced speech can be characterized by a lower-order predictor than voiced speech.

4. Low-Bit-Rate Vocoder

We were able to achieve a very simple variable-bit-rate system using the "Equal-Area" scheme, whose average bit rate for continuously read speech was about 1400 b/s. The tables were computed from histogram data from 7 speakers. Speech was recorded through a handset, and pre-emphasized by a fixed analog pre-emphasis. It was found that these fixed tables could be used for any speaker, but that the recording conditions had to be the same as those used for the histograms.

A bit rate of 1400 b/s was achieved using the following algorithm:

(a) For each frame, send a 2-bit code to indicate silence, buzz, or hiss.

(b) For silence frames, send no more bits.

(c) For unvoiced frames, send 5, 5, 4, 3, 2, 1 bits for K's 0 through 5, plus 5 bits for energy.

(d) For voiced frames, send 4, 5, 4, 4, 3, 3, 3, 3, 3, 2 bits for K's 0 through 9, plus 7 bits for pitch and 5 for energy.

The program keeps track of how many bits were used, and prints out the bit rate every 10 sec. In addition, it computes the relative amounts of time spent in voiced, unvoiced, and silence frames. Due to stopgaps and short pauses, a minimum of 25 percent of the frames are silence, even when a person is reading a passage of continuous speech. These silences would usually not be sufficiently long to be sent as silence packets across the net, and therefore represent a considerable savings in bits for minimal programming effort. We found that, on the average, the unvoiced frames are about two-thirds as frequent as the voiced frames. If, for example, 25 percent of the frames were silence, 30 percent were hiss, and 45 percent were voiced, the resulting average bit rate would be 1450 b/s.

II. BANDWIDTH COMPRESSION WITHOUT PITCH EXTRACTION

1. Introduction

For many years, speech research workers have recognized the vulnerability of "conventional" speech analysis-synthesis to environmental factors. By conventional, we refer to systems for modeling vocal tract and excitation parameters separately. The difficulty is most sharply exemplified by a helicopter environment, where the blade noise frequency "captures" the pitch extractor.

As long ago as the late 1950's, ways around this problem were being proposed, with the most prominent being the voice-excited channel vocoder scheme. With the advent of LPC, new "pitchless" configurations became possible. The Adaptive Predictive Coding (APC) scheme advanced by Atal and Schroeder used pitch information but was much less dependent on it than standard LPC. Under ARPA auspices, SRI introduced RELP (Residual-Excited Linear Prediction) and Lincoln introduced VELP (Voice-Excited Linear Prediction). The major differences between these two related systems are: first, the baseband signal transmitted in VELP was derived by selectively filtering the speech input, rather than by filtering the residual error signal as in RELP; and second, VELP employed an adaptive spectral flattening filter, with parameters determined by LPC analysis of the distorted baseband signal.

2. Unified Description of VELP and RELP

VELP and RELP block diagrams are given in Figs. I-4 and I-5, respectively. (The LPC analysis producing the K-parameters is not shown in the figures.) The only difference between the two systems is that RELP includes an additional linear filter prior to the baseband-selection filter. This inverse vocal-tract filter generally will not eliminate any harmonics present in the speech, but will tend only to flatten the spectral envelope. Hence, in both systems the output of the baseband-selection filter will contain the same harmonics. If the distorters and spectral flatteners properly perform their functions of generating all harmonics and flattening the spectral envelope, then the two systems should produce essentially the same excitation function spectra.

An adaptive spectral flattener was introduced for the VELP system as follows. The output of the distortion network (which currently incorporates, as in RELP, an asymmetrical half-wave rectifier and differencer) is analyzed by linear prediction, and fed through an inverse filter based on the derived linear predictor coefficients. The spectral flattening properties of such a filter have been discussed by Gray and Markel.* It is important to include a gain parameter in the flattening filter such that the energy in the generated excitation signal is proportional to the energy in the residual error signal of the original speech. Clearly, this same type of adaptive spectral flattening could be applied to RELP as well as to VELP.

In the VELP system, one can take advantage of the fact that a baseband of speech is being transmitted over the channel by incorporating this baseband signal into the output signal as indicated in Fig. I-6. In general, one would expect that the quality of a voice-excited vocoder would be enhanced significantly by the presence of a band of clear speech in the output. A corresponding technique can be employed in the RELP system by incorporating the transmitted residual error signal baseband into the excitation signal as indicated in Fig. I-7. The resulting effect should be the same as that due to adding in the speech baseband in VELP.

*A. H. Gray, Jr. and J. D. Markel, IEEE Trans. Acoustics, Speech, and Signal Processing ASSP-22, 207-217 (1974).

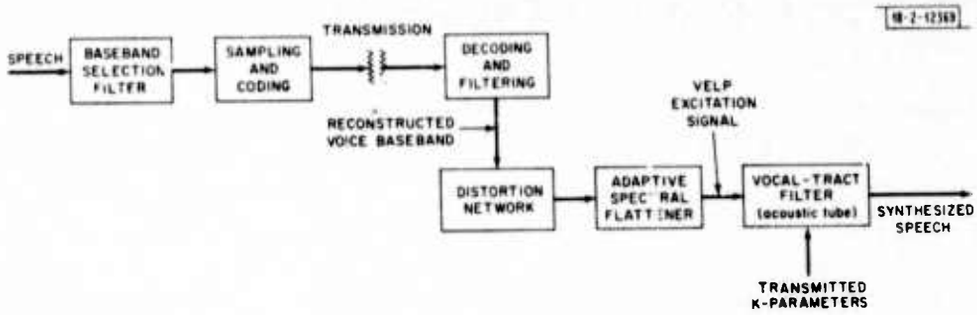


Fig. I-4. VELP vocoder with spectral flattening.

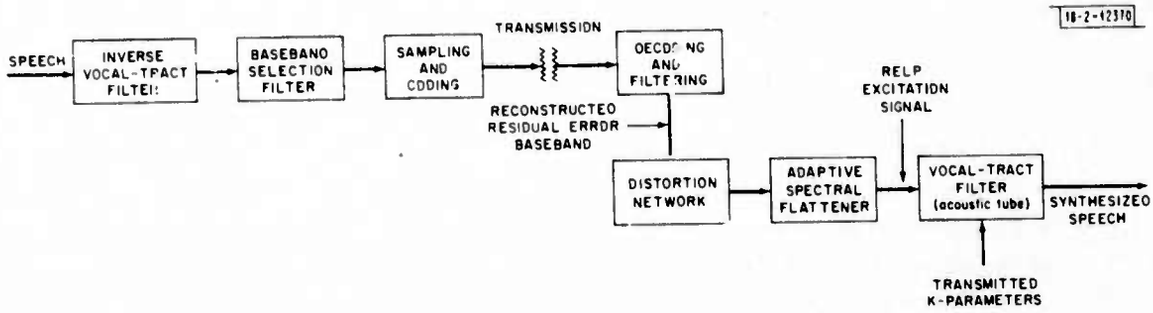


Fig. I-5. RELP vocoder with spectral flattening.

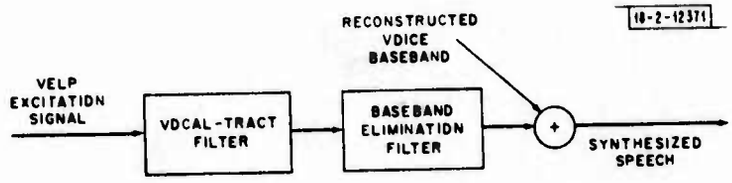


Fig. I-6. Combination of voice baseband with synthesizer output in VELP.

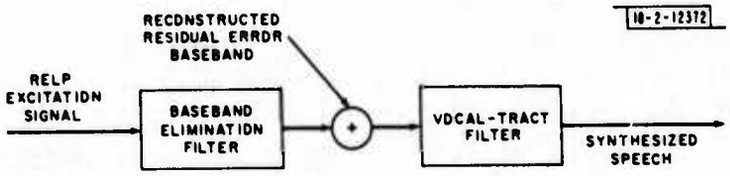


Fig. I-7. Combination of residual baseband with synthesizer input in RELP.

The above paragraphs have emphasized the inherent similarities between a residual-excited and a voice-excited system. The only important difference is the additional inverse filter in RELP, which costs extra processing time but might conceivably yield some advantage by providing a more equalized (in amplitude) set of baseband harmonics to the distorter and flattener, and perhaps ease their task of generating a spectrally flattened excitation signal.

3. Results to Date

VELP and RELP systems, both with and without adaptive spectral flattening, have been implemented in real time on the fast digital processor (FDP) at Lincoln. The basic system parameters, common to both VELP and RELP, are: sampling interval, 130 μsec; window size, 150 samples (Hamming); frame rate, ~60/sec; baseband-selection filter, 192 to 577 Hz; linear prediction order, 10 for both speech analysis and excitation signal flattening. When the LPC flattener is not used, the proper gain is achieved by requiring that the energy out of the distortion network be proportional to the energy in the speech residual error. Most of the work to date has been with uncoded systems in an effort to focus on such issues as the effect of the LPC flattener. The re-introduction of the baseband signal (see Figs. I-6 and I-7) so far has been investigated only in a preliminary fashion, and only in VELP.

The following observations have been made:

(a) The adaptive linear predictive spectral flattener seems to produce significant improvements in both RELP and VELP, in comparison to results obtained using just the rectifier and differencer. With LPC flattening, the output speech is smoother and clearer, and the observed excitation signal spectral cross sections are significantly flatter.

(b) It appears that both systems could benefit from more (i.e., higher than 10th order) rather than less flattening of the excitation signal. This conclusion was arrived at by taking pictures of spectral cross sections at the output of the LPC flattener, and noting that these spectra were generally not as flat as the corresponding spectra of the speech residual error signal.

(c) The excitation signal in RELP, after adaptive flattening, seems to be flatter than in VELP, apparently due to the fact that the baseband signal in RELP has been preflattened by the inverse vocal-tract filter.

(d) The two systems (with LPC flattening) sound different, but it seems hard to judge which is preferable. VELP has occasional energy irregularities due to a large (not well-flattened) excitation signal harmonic coinciding with a sharp formant. On the other hand, RELP has a generally somewhat rougher quality.

(e) Initial experiments with VELP indicate that the re-incorporation of the baseband signal (Fig. I-6) significantly enhances the output speech. Corresponding results are to be expected in RELP.

Work is currently proceeding to investigate further some of the issues raised by these observations and to develop fully coded systems. The hope is that a high-quality system can be developed at about 6 kb/s, which will operate well in environments (e.g., where a carbon microphone is used) where pitch-excited systems become very degraded.

1. SPEECH UNDERSTANDING SYSTEMS

The previous SATS described plans for the experimental evaluation of the Lincoln mid-term speech understanding system. Although work on speech understanding systems was terminated at the end of FY 74, some work on that experiment and the analysis of its results were carried out between the writing of the previous SATS and the end of the program. This section briefly reviews the experimental conditions, describes the results, and states some conclusions which we have drawn from those results and our experience with the system.

The speakers were seated before a CRT terminal in the TX-2 computer room. Fan and blower noises from various pieces of equipment were present, but there were no sharp clacking noises from teletype-like devices. The speaker used a noise-canceling headset-mounted microphone (Roanwell P/N 116830-690).

Six male speakers were used. Three had considerable experience with the system and we expected them to have at least moderate success. The other three speakers had known of the system, but had little if any experience. These latter three speakers had speech patterns similar enough to the known speakers so that we anticipated similar performance. It is important that a speaker have a natural even cadence throughout a sentence, with no tendency to let the end of the sentence trail off.

Each speaker was told that the sentences were to be formulated from a grammar that dealt with retrieval and analysis of speech data. He was shown a list of the 150 vocabulary words and was assured of all pronunciations. The speaker then was shown a chart which described a set of sentences acceptable to the system's grammar.

The session was to consist of the presentation of a series of 25 such charts. The speaker's task was to formulate a legal sentence from the chart and speak the sentence into the microphone. A sentence was formulated by choosing a path and selecting one of the alternate words at each node. Some charts were simple and allowed only a dozen or so sentences. Most of the charts were more complex and allowed up to thousands of sentences. The charts spanned the entire grammar, and an attempt was made to uniformly sample the grammar. After formulating and speaking the sentence, the speaker saw the analysis produced by the system.

About a week later, the speaker returned to the TX-2 room and recorded a random list of the 25 sentences he had formulated earlier. For each subject, we then had 50 sentences: 25 spoken in the real system context, and the same 25 recorded with no response from the system.

Before discussing the results, we must make two points clear. First, the linguistic analysis used two matrices to score words – one for vowels and one for consonants. The matrices used throughout the experiment were computer-generated from the analysis of some 113 sentences. None of the data collected in this experiment was used to generate these matrices, but some of the data came from speakers who spoke some of the 113 sentences. Second, a table of speaker-dependent vowel formant positions was used in the front-end processing. This table had already existed for one of our speakers. We decided that these values would not need to be changed for the other speakers, and we used this one table for all speakers.

Table I-1 shows the overall results for the six speakers. Note that one speaker was not available for the repetition of his sentences. For both sets of sentences, about 50 percent were correctly recognized by the system and about 20 percent were unanalyzable or nonsense. In this table, "correct" means the lexical output of the system matched the speech word-for-word. "Bad" means that the lexical output bore little if any resemblance to the speech, or that for reasons of time or space the analysis was incomplete. "Partial" means that a substantial part of the sentences was correctly recognized, but that one or more words was in error.

TABLE I-1

RESULTS OF THE EXPERIMENTAL EVALUATION
OF THE LINCOLN MID-TERM SPEECH UNDERSTANDING SYSTEM
(C = correct, P = partial, B = bad)

	Formulated			Read		
Speaker	C	P	B	C	P	B
1	9	8	8	12	9	4
2	13	9	3	12	11	2
3	14	7	4	14	9	2
4	11	6	5	14	6	5
5	13	7	5	13	5	7
6	11	10	4	–	–	–
Total	71	50	29	65	40	20
Percent	(47)	(33)	(19)	(52)	(32)	(16)

The following example sentences, which are typical of those produced by all speakers, illustrate some partially correct sentences. In the examples, the underlined words are those which were incorrectly identified by the system; the words in parentheses show the system's interpretation of the speech corresponding to the underlined words.

Find the liquids for the current entries (utterance ninety three).
Give-me all sentence information (the).
Print the voiceless plosive on the scope (stop from entry ten).
Move the tape to the current file (fourth).
List all the affricates on the xerox (all).
Move to slot eighty four (three).
Search for the sonorants (stops).
Retrieve the front vowels (fourth front).
Redisplay all matches of this file (the waveforms).
Find the initial consonant in the current slot (first).
Skip to the next utterance on tape unit two (eight).
Skip to the next utterance on tape unit two (slot).
Pickout the voiced fricatives in the current entry (utterance).
Compute the maximum pitch in the first plosive (current glide).
Compute the maximum pitch in the first plosive (stop).
Display the formants for the shortest word (sonorants).
Delete those shorter than fifty seven milliseconds (eighteen).
Try-to-find all the phonemic labels from the computer (drum).

Of the various factors contributing to sentence failure, the most significant are missing and extra syllables. We feel that the possibilities for correcting these problems at the linguistic level are not good, and that efforts in this area should focus on the front end.

The capability achieved by the mid-term system does not represent anything like the limit of performance obtainable with a system of the type we have been investigating. We estimate that another round of tuning, taking into account our experience to date, but without extending the front end, would yield another 10 to 15 percent more correct sentences. We anticipate that the incremental gain from optimal use of pragmatic or discourse information would be somewhat less. Further improvement then must come from extension of the front-end processing techniques.

In extending the front end, we first would make use of durational information which is now largely ignored, and provide confidence measures to the word matcher. This latter step looks as if it could be very helpful in correcting the "bad" sentences. Beyond those steps, we would explore the use of prosodic cues in a verification mode, and try to improve on the use of formant motion cues by taking coarticulation phenomena into account. This processing looks expensive, and would thus be done only in verifying already attractive hypotheses.

II. ACOUSTIC CONVOLVERS

A. INTRODUCTION

The Lincoln Laboratory effort on acoustic convolvers for packet radios is motivated by a desire to increase the instantaneous bandwidth and correlation gain of continuously variable, pseudo-random, phase-shift-keyed modems without paying the price of long acquisition time.

The base-line receiver subsystem which we are considering has an instantaneous bandwidth capacity of 100 MHz and a data-rate capacity of 10^5 b/s. Four convolvers, each with a convolution interval between 10 to 15 μsec, are connected sequentially with delay lines to provide as much as 36 dB of correlation gain during the acquisition mode. Once the system is synchronized, only a single convolver might be used, which provides a maximum available correlation gain of 1000 (or 30 dB) at a bit rate of 100 kHz. Under severe channel interference conditions, the bit rate might be reduced to 25 kHz in order to increase the bit correlation gain. Ultimately we might also provide, for a virtually noise-free environment, provision for an increased data rate of 10^6 b/s. In this instance, special shorter convolvers would be implemented, each with a convolution interval of about 1 μsec and with a correlation gain of 20 dB.

A design goal for the base-line receiver might be to provide synchronous phase lock of pseudo-random, phase-shift-keyed code which is spread over a bandwidth of 100 MHz. A short preamble of no more than 20 bits of pseudo-random code might be used to achieve synchronism when the arrival time of an incoming signal is known to within a few milliseconds. This foreknowledge is desirable in order to keep the number of false alarms at the receiver within reasonable bounds.

Four convolvers could be used in unison to provide a correlation gain of 36 dB during the acquisition phase. Once synchronism is obtained, one convolver could be used to decode the message, and the remaining convolvers could be used to verify the accuracy of the decoded text and to maintain time and phase synchronism of the receiver with the incoming signal. The receiver could be designed to have fewer than one false alarm per millisecond during the acquisition phase, and we have conceived a scheme in which a receiver can recover from an occasional false alarm without jeopardizing the acquisition of an incoming message packet.

We expect to place no requirements or constraints on the selected code structure, or on the modulation method which is used. Convolvers should work equally well with biphase, quadrature-phase, differential-phase, frequency-modulated modems, and with sequential complementary codes.

B. PROBABILITY OF ERROR AND FALSE ALARM

The probability of error is plotted in Fig. II-1 as a function of signal-to-noise ratio (SNR) at the receiver output for several different cases. Suppose a probability of error of 10^{-5} is chosen during synchronous reception. There is a chance that one error occurs in every 100 packets of 1000 bits. A SNR after correlation of 9.5 dB is required for the case where the receiver is synchronized to the incoming signal and where a narrow sample is taken at the peak of the correlation spike. However, it is difficult to achieve accurate timing, partly because we intend to use wideband signals in which a correlation pulse width is about 5 nsec. Figure II-2 shows the error probability as a function of timing gate error T_e, which is normalized with respect to the chip time interval T_c. Note that an error of 1/4 of a chip interval degrades the probability of

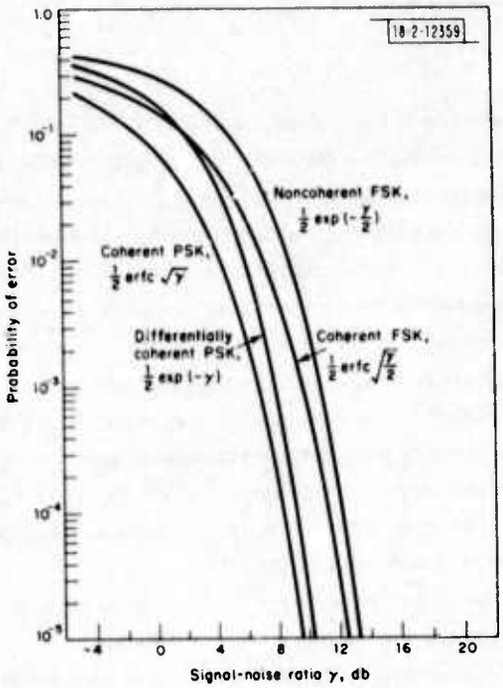


Fig. II-1. Probability of bit error vs SNR ($\gamma = E/N_O$). [Reprinted with permission from Communications Systems and Techniques by M. Schwartz, W. R. Bennet and S. Stein (McGraw-Hill, New York, 1966), p. 299.]

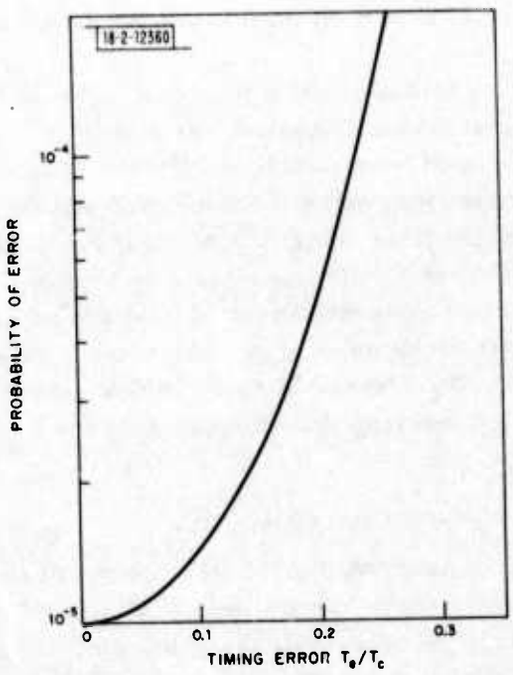


Fig. II-2. Error probability vs timing-gate error for band-limited convolver. It is assumed that circuit bandwidth matches input signal bandwidth. Error probability increases as timing gate is displaced from correlation peak by time T_e. Probability of error is plotted as a function of timing error T_e, normalized to chip time T_c, for coherent PSK. Limited bandwidth provides correlation peak which is smoothed, and error probability changes little for $T_e/T_c < 0.1$.

error from 10^{-5} to 10^{-4}. Control of bit timing might be accomplished by slaving the bit and chip interval in the receiver to a local oscillator (LO). If the LO is phase-locked to the incoming signal, it should be possible to achieve accurate sampling-gate timing.

Before the system is synchronized and while it is waiting for preamble, it is desirable to have, on the average, no more than one false alarm during a millisecond. (The technique by which the system recovers from a false alarm is described in Sec. E below.) Since we expect to receive 10^5 chips per millisecond, a false-alarm probability of 10^{-6} is desirable. By using four convolvers during acquisition, an additional 6 dB of correlation gain is available and an output SNR of 15.5 dB is expected. Figure II-3 shows that the probability of detecting a preamble is greater than 0.999 for the situation where the timing gate is wide open. Once a signal is detected, the timing gate can be narrowed to about a chip interval, thus aiding in noise rejection. It is then possible to begin the process of time-locking, and ultimately phase-locking the receiver LO to the incoming signal. The stability of the receiver LO is expected to be 1 part in 10^6.

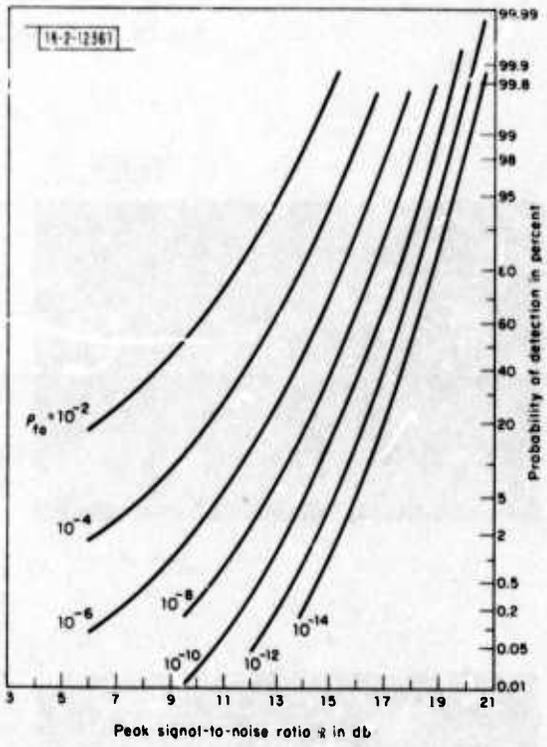


Fig. II-3. Probability of detection with respect to SNR for various false-alarm probabilities (P_{fa}). $\Re \equiv 2\ E/N_o$, hence the 3-dB difference. Curves assume envelope detection. [Reprinted with permission from Radar Detection by J. V. DiFranco and W. L. Rubin (Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1968), p. 308.]

C. CONVOLVER PERFORMANCE AND DESIGN CONSIDERATION

The acoustoelectric convolver is a three-port device shown in Fig. II-4. In operation, RF signals are fed to ports 1 and 2 and the convolution of the two input signals appears at port 3. If one of the input signals is a time-reversed replica of the other, the signal present at port 3 is the autocorrelation of the input signal. Under this condition, the convolver is equivalent to an ideal matched filter with an impulse response given by the time-reversed input. By changing this reference input, the convolver becomes a programmable matched filter.

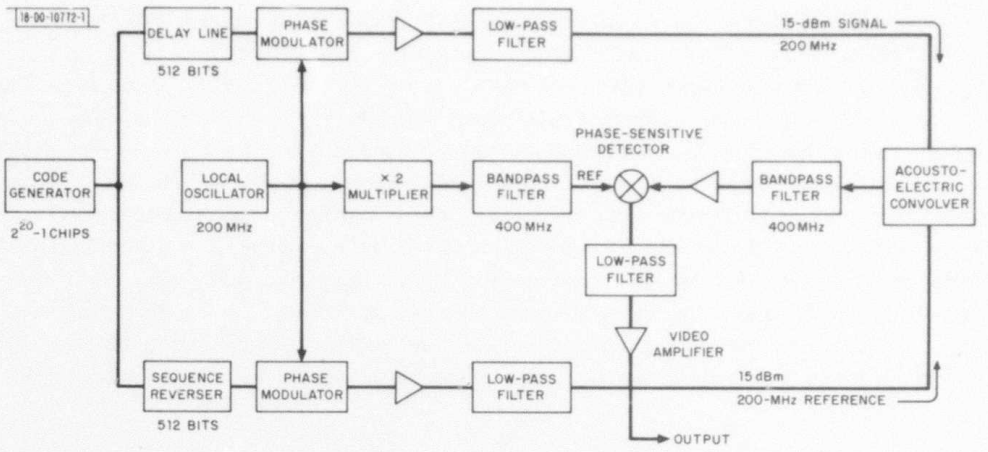


Fig. II-4. Convolver test circuit.

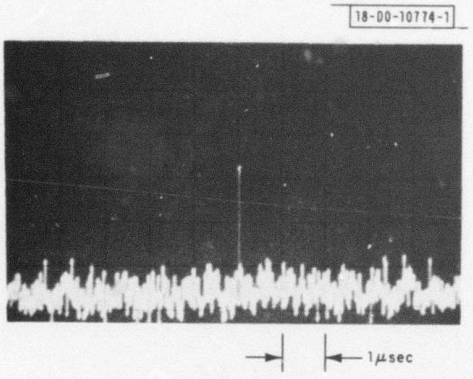


Fig. II-5. Convolver output. (a) Repeated code; (b) running code. Input: pseudo-random PSK code, f_o = 200 MHz, W = 67 MHz, 30 nsec/chip, 233 chips/bit, 7 μsec/bit.

An acoustoelectric convolver was assembled and placed in a special test circuit in Fig. II-4. This test convolver operated at a center frequency of 200 MHz, with a bandwidth of 67 MHz and a convolution interval of 7 μsec. The test circuit consists of a 20-bit maximal length, pseudo-random shift register which generates a code that repeats every 1,048,575 bits. The output of the code is divided into 512-bit sub-sequences. One of the sub-sequences is reversed in time by the digital-bit reverser. The digital-bit delay is used to delay the unreversed bit stream to match the delay through the bit reverser. Both the delayed and reversed sub-sequences are fed to separate 0/180° phase modulators which modulate a common RF source. The output of the phase modulators is fed to the convolver inputs. The output of the convolver is amplified and coherently detected. Since the convolver output frequency is twice the frequency of the phase-modulated signal, a carrier at twice the RF frequency is needed for coherent detection. This carrier is generated by the doubler and fed to the mixer. A phase control assures that the doubled frequency will be in the correct phase for maximum output. The output from the mixer is the coherently detected signal. This signal is monitored on an oscilloscope.

In this preliminary experiment, a pseudo-random code sequence with a cycle interval of 4500 bits was decoded with the convolver. In this instance, a LO frequency of 200 MHz was used, the number of chips per bit was 233, the chip interval was 30 nsec, and the bit duration time was 7 μsec. The particular code was selected for long cycle time, and no attempt was made to minimize the time sidelobes. The output amplitude of the convolver is shown in Fig. II-5(a-b), and Fig. II-6(a-b) shows the ideal calculated time sidelobes. Figure II-5(a) shows the correlation signal for a single-bit sequence, and (b) shows the correlation impulse

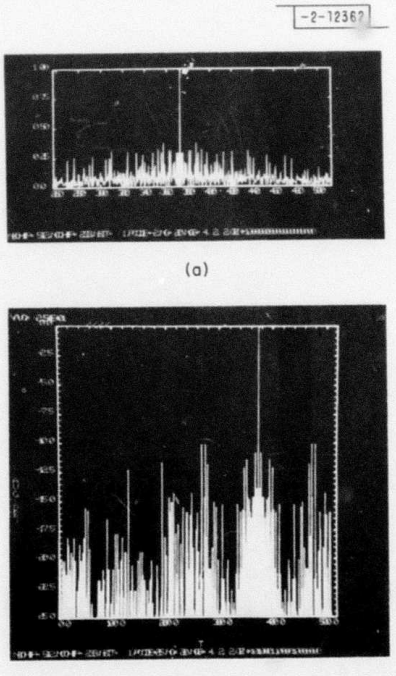


Fig. II-6. Theoretical convolver output for one bit of continuously running code: (a) shows time waveform; (b) shows sidelobe structure on decibel scale.

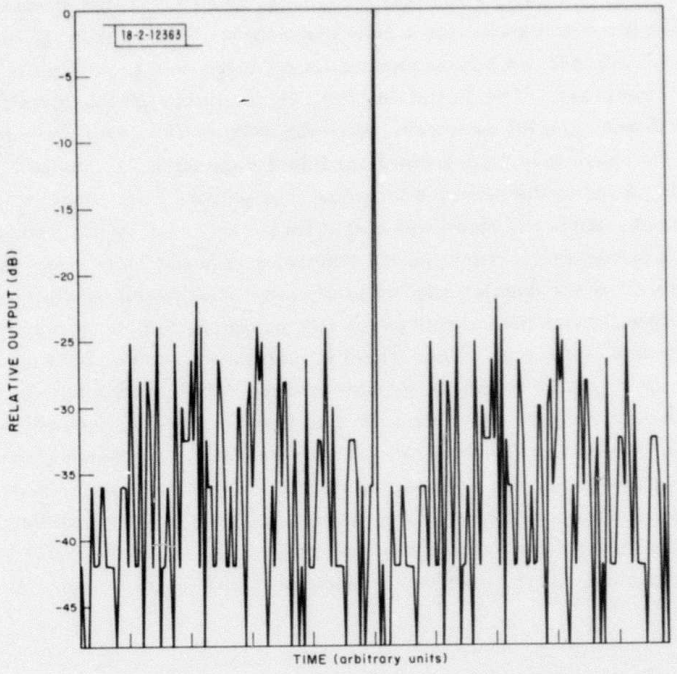


Fig. II-7. Calculated correlation characteristics for special low time-sidelobe code. Code is repeated every 127 chips. Calculation is for 250 chips.

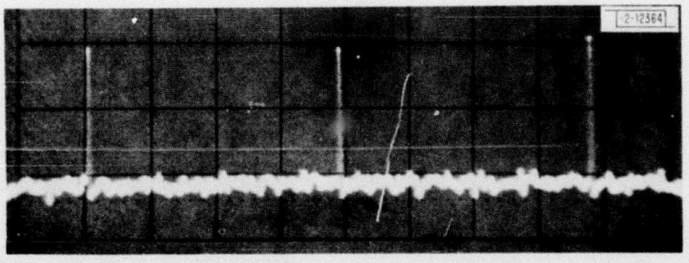


Fig. II-8. Convolver output for low sidelobe code.

for a continuously varying code. The correlation impulses for the continuously changing code are overlaid in Fig. II-5(b). Notice the change of scale; here, 1 cm is 0.1 μsec. In this experiment, the width of the correlation impulse and the rms sidelobes are close to the ideal values.

A special code sequence was selected which is known to have theoretical time sidelobes which are 23 dB below the correlation pulse (see Fig. II-7). The convolver outputs for this code are shown in Fig. II-8. Notice the well-defined correlation impulses. Also notice time sidelobes which are approximately 23 dB below the main signal. Again, these sidelobes are artifacts of the code, and not of the convolver. The performance of the convolver consequently appears adequate for this code.

Phase and amplitude errors, and spurious signals in convolvers should cause sidelobes which are no greater than the tolerable noise floor at the output. This corresponds to approximately −90 dBm. One likely cause of time sidelobes is the reflection of the reference signal from the input transducer. This reflection convolves with the subsequent reference signal to produce a hash level which is expected to be, in our convolver, at a level of approximately −70 dBm. A simple correction of this problem would be to reduce the acoustic reflection coefficient of the input transducer from its current expected value of −20 to −30 dB. This can be done by decoupling the transducer from the electrical drive circuitry, or by using a dual-beam convolver in which two channels are driven in phase quadrature and in which the reflections at the transducers are caused to interfere destructively.

Calculations show that a dual-beam system requires convolvers in which the Si to $LiNbO_3$ gap should be accurate to ±50 Å. This accuracy lies beyond the state of the art, and therefore is not a viable correction for the reflection.

If the input transducer is decoupled from its drive circuitry by an additional 5 dB, then the acoustic reflection coefficient is 30 dB, which is adequate for our purposes. However, this causes an increase in insertion loss, and the input signal level must be increased from 20 to 25 dBm if we are to maintain at least a 50-dB dynamic range. This increased power level may be undesirable for some applications.

A tentative design goal for the acoustoelectric convolver is a center frequency of 300 MHz, a bandwidth of 100 MHz, a drive input power of 20 dBm, a dynamic range of 50 dB, spurious levels 30 dB below the main signal, and a duration time of 10 μsec or more.

An assembly procedure has been perfected for acoustoelectric convolvers. A diagram of the assembly is presented in Fig. II-9, and photographs are shown in Figs. II-10 and II-11. A special assembly jig was built to implement this assembly procedure. The silicon strip is placed in a molded RTV gel and the $LiNbO_3$ plate is moved into position over the Si. The structure is clamped together with two screws, which exert sufficient pressure on the gel to cause it to deform around the Si and capture it in position. The Si is held off the $LiNbO_3$ surface with 3000-Å-high pseudo-randomly distributed posts that are left standing on the $LiNbO_3$ surface. The acoustic beam propagates through these posts with minimal loss.

We are currently evaluating the convolver described by Defranould at the 1974 Ultrasonics Conference.* In this structure, the acoustic energy in a 100-wavelength-wide beam is compressed into a 7-wavelength-wide waveguide with the aid of strip couplers. If the input and reference power levels are 30 dBm, then the energy in the strip waveguide is 30 dB above the

* P. H. Defranould and C. Maerfeld, "Acoustic Convolver Using Multistrip Beamwidth Compressors," Ultrasonics Symposium Proceedings, 11-14 November 1974, pp. 224-227, IEEE CHO896-1SU.

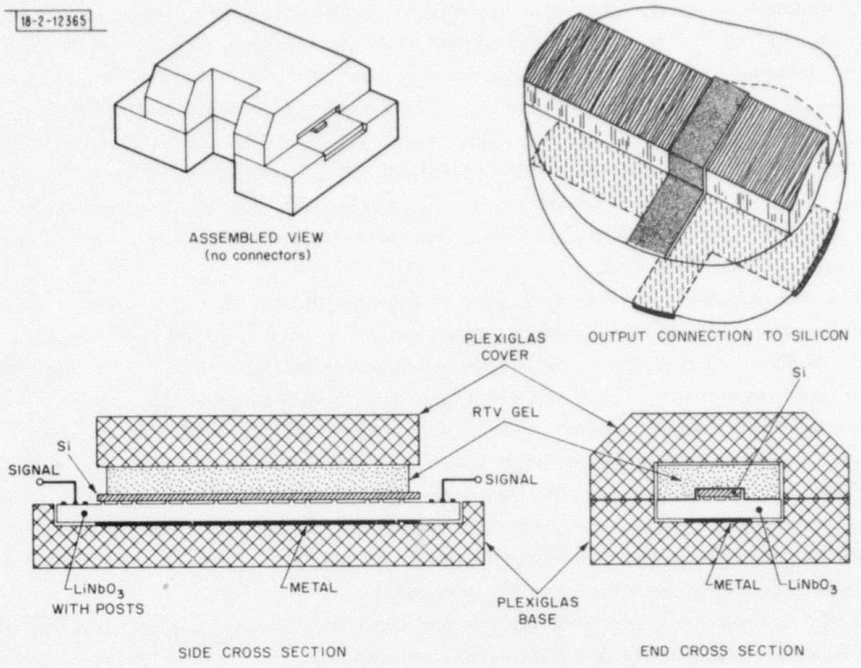


Fig. II-9. Cross-section view of convolver.

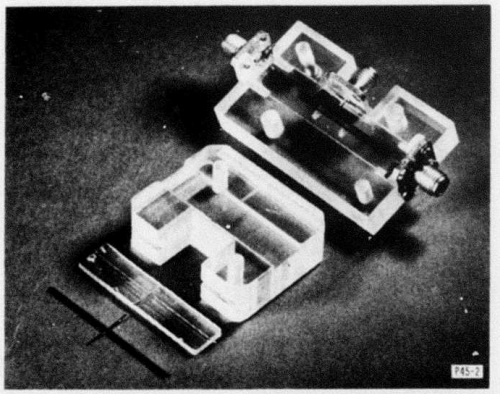


Fig. II-10. Photograph of convolver parts.

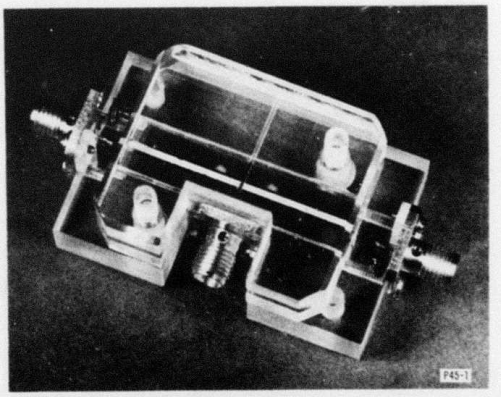


Fig. II-11. Assembled convolver.

nonlinear acoustic limit. This creates a highly nonlinear medium in the waveguide, and efficient convolution is obtained. This convolver has the desirable feature that it is made only with monolithic metal overlays and does not require Si strip. However, one undesirable feature is that the power required to drive the acoustic medium nonlinear is about 10 dB above the requirements for acoustoelectric convolvers with similar performance.

D. MEMORY CORRELATORS

Lincoln Laboratory has been actively engaged in perfecting acoustoelectric memory correlators. We have been investigating several memory mechanisms including free electrons on piezoelectric surfaces, trap states in the surface of Si, Schottky barrier diodes on Si, and metal-oxide-metal tunneling devices. The methods which appear most promising for spread spectrum receivers are Schottky barrier diodes and silicon trapping states. The memory correlator consists of a $LiNbO_3$ delay line, and a Si wafer, the surface of which is either especially treated to enhance trap characteristics or is covered with a dense carpet of free-standing Schottky barrier diodes. The signal is introduced on $LiNbO_3$, and when it is beneath the Si, a sharp impulse is applied to the Si. This causes an instantaneous storage of the free charge appearing on the Si surface due to the surface wave on the $LiNbO_3$. Thus, an image charge is retained of the acoustic signal on the Si surface for as long as several milliseconds. This image charge can be used as a reference against which subsequent signals correlate.

Memory correlators require one bit time-interval for loading the reference and a subsequent bit-interval for correlating with that reference. Consequently, for running codes it is necessary to provide two devices: one performs the current correlation, while the other stores the next reference.

The memory correlator is a true programmable matched filter, where the output correlation function, at the fundamental frequency, has a duration of twice the chip interval. This is in contrast to the convolver, in which the correlation function is time-compressed by a factor of 2. Also, in the memory correlator it is not necessary to time-reverse the reference. Because the reference is stored in a memory correlator, the clock at a receiver could precede the incoming signal by as much as the storage interval of several milliseconds.

The technology which is currently being developed for acoustoelectric convolvers is directly applicable to the realization of memory correlators. Consequently, we plan to choose the acoustoelectric version.

E. CONVOLVER CIRCUIT

Four convolvers are connected in series during signal acquisition in order to provide 36 dB of correlation gain. After the receiver is synchronized to the incoming signal, one convolver could be used to decode the text, and the remaining convolvers could be used to maintain sync and to verify the decoded message. The block diagram in Fig. II-12 shows the functional relationships of the convolvers with circuit elements. It is assumed that a signal packet arrives with a pseudo-random coded preamble. Each bit of the preamble has its own unique code, and the bits are numbered 1, 2, 3, etc., as shown in the top of the figure. The reverse code sequences for bits 1 through 4 are loaded into shift registers SR1 to SR4, while the shift register switches are in the horizontal position for four bit-intervals. The switches are thrown into the vertical position and the baseband code sequences circulate through each shift register. The

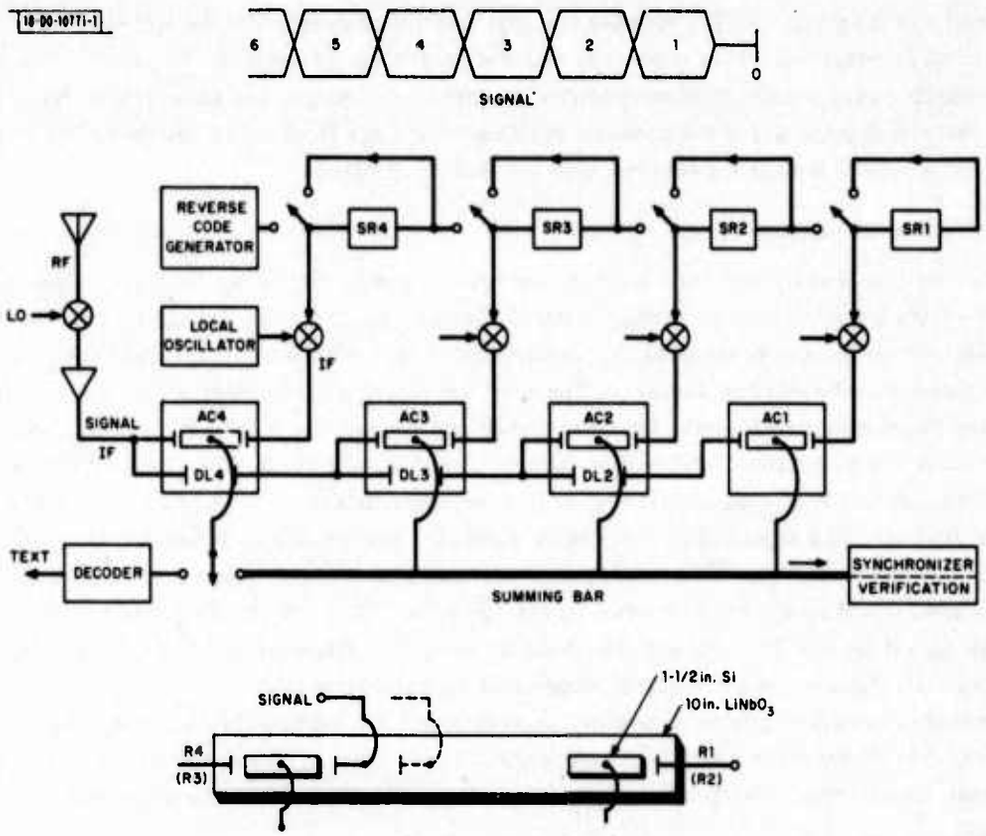


Fig. II-12. Convolver circuit layout.

code signals are translated to the intermediate frequency with mixers. These reference signals are entered at the right side of acoustic convolvers AC1 to AC4. Thus, the reference of bit 1 circulates through AC1, bit 2 through AC2, and so on.

The incoming packet is translated from RF to IF with a LO mixer. The incoming signal passes through delay lines DL4, DL3, and DL2, and bit 1 enters AC1 as bit 2 enters AC2, etc. Each delay is precisely 1 bit in duration. Consequently, four simultaneous correlation impulses enter the summing bar and trigger the synchronizing circuits. (At this instance, the switch of AC4 is thrown to the right.)

The detection of a correlation impulse turns on the reverse code generator, and the switches at the shift registers are thrown into the horizontal position. The code sequence for bit 5 enters CR4, the code sequence for bit 4 enters SR3, etc. Reverse reference signals for bit 5 enter the right side of AC4, for bit 4 enter AC3, and so on. Incoming bit 5 enters AC4 from the left, bit 4 enters AC3, etc. Consequently, a second correlation impulse emerges from the summing bar one bit-interval after the first. These correlation impulses should provide an adequate signal for synchronizing the timing and phase of the receiver to the incoming signal.

If this initial impulse is a false alarm, then the second impulse one bit-interval later will be missing. In this event, the switches to the shift registers are thrown into the vertical position and the reverse code generator is turned off. Now reverse code references for bits 2 through 5

circulate through AC1 through AC4. A correlation impulse is obtained when bits 2 through 5 register with their reverse references in the convolvers. Thus, a false alarm has the effect of dropping the first bit from the preamble.

If PSK is used, then the output of AC4 is switched to the decoder once time and phase synchronization are obtained. The decoded text could be entered between SR4 and SR3, and the references entering AC1 through AC3 would correspond to the first estimate of the text. If the decoded message is correct, then the correlation impulse entering the verification circuit has a relative amplitude of 3. If an error is made, this amplitude drops to 1. The same probability of error is expected in the verification circuit as in the decoder.

If DPSK is used, the correlation impulses out of AC4 and AC3 are compared to decode the text. AC1 and AC2 could be used to provide verification and synchronization functions as before.

It is not practical to connect wide-band delay lines in series as shown in Fig. II-12, because an unacceptable amount of band compression is expected for signals passing through as many as seven acoustic transducers. A more desirable arrangement is to use parallel delay lines having a progressive delay corresponding to 1, 2, and 3 bits. The convolver layout at the bottom of the figure shows how this might be done. Two large plates of $LiNbO_3$, perhaps 8 inches long, would be used. The two devices would be identical except that the input transducers for the signal would be located in such a manner as to provide the desired differential delay. For example, if the signal enters the solid input terminal, then the delay from the transducer to the convolver structure on the right is equal to three bit-intervals. If the signal enters the dotted input, the delay to the convolver on the left is one bit-interval and to the right is two bit-intervals. The output of SR4, which provides references R4, enters the terminal on the left and R1 enters from the right. The second plate with the input transducer at the dotted terminal has R3 from the left and R2 from the right.

During the acquisition mode, it is necessary for the correlation impulses to line up accurately in time if the full correlation gain is to be obtained. This requires that the delay lines be accurate to one part in 10^5. Since $LiNbO_3$'s temperature sensitivity is 100 ppm per degree centigrade, it is necessary to keep the $LiNbO_3$ substrates at their design temperature within 1/20th of a degree centigrade. Alternatively, it might be possible to sense the variation of delay and compensate for it by introducing appropriate phase delays to the reference signal. We are also studying the possibility of building a convolver with a 40-μsec convolution interval. A much greater variation in temperature could be tolerated by convolvers which contain the full 4-bit convolution interval.

F. CONCLUSIONS AND RECOMMENDATIONS

The acoustic convolver has been used successfully to provide programmable matched filter functions for spread spectrum signals. The tests and computer simulations suggest that time sidelobes due to subsets of shift register have an rms level equal to $\sqrt{N}$, where N = number of chips/bit. However, peak values often approach −10 dB below the main correlation pulse level. This appears to be so in the case for N as large as 1000. Considerable care should be taken to suppress reflected acoustic reference signals, since they are likely to cause spurious signals. We are recommending that segmented convolvers be used during the acquisition mode. This implies the use of acoustic delay lines.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: ESD-TR-76-100 Vol-I	2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle): Information Processing Techniques Program, Volume I: Packet Speech/Acoustic Convolvers	5. TYPE OF REPORT & PERIOD COVERED: Semiannual Technical Summary, 1 Jul - 31 December 1974
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): Bernard Gold, Ernest Stern	8. CONTRACT OR GRANT NUMBER(s): F19628-73-C-0002; ARPA Order-2006
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Lincoln Laboratory, M.I.T., P.O. Box 73, Lexington, MA 02173	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: ARPA Order Nos. 2006 & 2929; Program Element Nos. 61101E & 62708E; Project Nos. 6P10 & 6T20
11. CONTROLLING OFFICE NAME AND ADDRESS: Advanced Research Projects Agency, 1400 Wilson Boulevard, Arlington, VA 22209	12. REPORT DATE: 31 December 1974
	13. NUMBER OF PAGES: 36
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office): Electronic Systems Division, Hanscom AFB, Bedford, MA 01731	15. SECURITY CLASS. (of this report): Unclassified
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

ARPA Order-2929

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

None

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

information processing techniques
packet speech
network speech compression
acoustic convolvers
speech understanding
digital speech transmission
ARPANET

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The Information Processing Techniques Program sponsored by DARPA at Lincoln Laboratory consists of three efforts: Packet Speech (Network Speech Compression), Acoustic Convolvers, and Airborne Command and Control. In this Semiannual Technical Summary, the first two areas are reported in Vol. I and the third in Vol. II. In addition, Vol. I contains a brief summary report on work in Speech Understanding completed in FY 1974.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

207650